Why GRPO Needs Normalization: A Local-Curvature Perspective on Adaptive Gradients

Cheng Ge*
Department of Aeronautics and Astronautics
MIT
gec_mike@mit.edu

Caitlyn Heqi Yin*
Department of Statistics
University of Wisconsin–Madison
hyin66@wisc.edu

Hao Liang†
Department of Informatics
King's College London
hao.liang@kcl.ac.uk

Jiawei Zhang†
Department of Computer Sciences
University of Wisconsin–Madison
jzhang2924@wisc.edu

Abstract

Reinforcement learning (RL) has become a key driver of language model reasoning. Among RL algorithms, Group Relative Policy Optimization (GRPO) is the de facto standard, avoiding the need for a critic by using per-prompt baselines and variance normalization. Yet, the role of normalization remains unclear. In this work, we provide an explanation through the lens of local curvature of the sequence-level policy gradient. We show that standard deviation normalization implements an adaptive gradient, improving convergence when curvature varies across prompts and across iterations. Furthermore, empirical studies on synthetic tasks and GSM8K confirm that normalization consistently improves stability and convergence, especially on harder problems with high reward variance. By establishing the connection between normalization and adaptive gradient, we provide a theoretical foundation for the empirical success of GRPO and offers broader insights into the design of critic-free RL algorithms for LLM training.

1 Introduction

Large language models (LLMs) have recently exhibited striking gains in multi-step reasoning, particularly when a lightweight reinforcement-learning (RL) stage is applied on top of a strong, pretrained and instruction-tuned base model. Among the many post-training recipes, Group Relative Policy Optimization (GRPO) has emerged as a practical, critic-free alternative that has powered some of the most visible reasoning systems [26], where it consistently improves solution accuracy under tight compute budgets.

While Proximal Policy Optimization (PPO) [25] remains a popular default in RLHF pipelines, it couples the policy with a learned value-function critic (and often GAE [24]). This increases memory footprint and implementation complexity, since the environment is a pretrained LLM and rewards arrive only at the end of whole sequences. This has renewed interest in critic-free policy-gradient methods that operate at the sequence level, such as REINFORCE-style method (ReMax [13] and its multi-sample extension RLOO [2]), which often match or outperform PPO for LLM alignment while being simpler and lighter-weight.

*Author order is alphabetical denoting equal contributions.
†Co-last authors

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

Classical REINFORCE [31] reduces gradient variance by subtracting a baseline (e.g., a running mean reward), yielding an unbiased estimator with lower variance [7]. This is textbook policy-gradient variance reduction. GRPO goes one step further: for each prompt, it samples multiple responses from the current policy, computes the group mean reward as a baseline, and normalizes each response's update by the within-prompt reward standard deviation, which effectively uses a per-prompt z-scored advantage [26]. Empirically, this simple normalization has been repeatedly observed to stabilize optimization and improve sample-efficiency in LLM RL [11]. Yet, the underlying mechanism of this normalization step has not been theoretically clarified.

What, exactly, does normalization do? In this work, we provide a principled explanation of why GRPO benefits from it. Our key insight is that the reward variance of each question serves as an estimate of the local Lipschitz constant of the policy gradient, i.e., its local curvature. Standard deviation normalization therefore acts as an adaptive gradient mechanism, scaling updates according to the smoothness of each question. In effect, it sets step sizes proportional to the inverse of the local curvature, which improves both stability and convergence when curvature varies significantly across questions and iterations. This perspective explains why GRPO consistently outperforms unnormalized policy gradient methods.

We provide both empirical and theoretical evidence to support this explanation. Our contributions can be summarized as follows:

- **Theoretical results on sequence-level bandit.** We show that, under a standard sequence-level bandit formulation, the per-prompt reward variance controls the local Lipschitz constant of the prompt-specific policy-gradient. Consequently, REINFORCE applies a single learning rate across heterogeneous prompts, whereas GRPO's variance normalization implements a prompt-wise and iteration-wise adaptive step size aligned with the local curvature, allowing adaptation across both prompts and iterations. Under an intuitive assumption of orthogonal representation, which guarantees that the gradients associated with different questions are orthogonal, we prove that GRPO attains provably faster convergence than unnormalized REINFORCE.
- **Empirical validation.** We validate our theory through (i) orthogonality checks of question representations, (ii) comparisons of three normalization strategies on GSM8K across difficulty levels, and (iii) synthetic tasks varying reward variance. Standard deviation normalization consistently improves stability and convergence under high variance, supporting our curvature-based explanation.

Our results explain why GRPO needs normalization: it is not only variance reduction, but a principled adaptive gradient mechanism that adapts learning to per-prompt curvature. Our work provides a theoretical foundation for the empirical success of GRPO and offers broader insights into the design of critic-free RL algorithms for LLM training.

1.1 Related works

REINFORCE-style PG methods. ReMax proposes a simple sequence-level REINFORCE objective for LLM alignment with strong performance and minimal complexity [13]. RLOO extends this by sampling multiple responses per prompt and using a leave-one-out baseline to further reduce variance [2]. REINFORCE++ continues this line, emphasizing simplicity and efficiency at scale [9].

GRPO and its variants. GRPO has become the default in state-of-the-art reasoning systems, combining a per-prompt baseline with within-prompt standard-deviation normalization [26]. Large-scale systems work (e.g., DAPO) has consolidated GRPO-style training across diverse tasks and compute regimes [33]. Related analyses examine design choices in normalization and sampling [17].

Emerging theory for GRPO. Recent studies analyze what GRPO optimizes and how it behaves in on- and off-policy regimes [20], its implicit alignment objective [29], and trajectory-corrected variants with convergence guarantees [22]. Other work highlights a trade-off between normalization and calibration, showing that removing the std term can improve probability calibration at the cost of optimization speed [3]. We contribute a new perspective: interpreting the std term as an adaptive gradient mechanism tied to local curvature, thereby unifying disparate empirical observations.

RLVR. Reinforcement learning with verifiable rewards (RLVR) has emerged as an effective paradigm for reasoning-intensive domains. Unlike RLHF, which relies on a learned reward model,

RLVR uses deterministic, verifiable rewards such as correctness checks [12, 8, 28, 30]. This avoids reward-model bias and simplifies training, while scaling effectively with compute and dataset size. Strong results have been reported on GSM8K, MATH, Omni-MATH, and FormalMATH [35, 16]. In this paper, we study GRPO in the RLVR setting, where deterministic rewards enable sharper theoretical analysis of normalization and its role in adaptive gradient updates.

2 Preliminaries and problem settings

We begin by formalizing the RLVR framework for LLM training and reviewing the GRPO algorithm. Then, we present our choice of policy parametrization along with the update details.

Notation. For a finite set $\mathcal{X}$, we use $\Delta(\mathcal{X})$ to denote the set of probability distributions over $\mathcal{X}$. By default, all vectors are column vectors. Unless specified, $\|\cdot\| = \|\cdot\|_2$ will always denote the standard 2-norm for vectors, and the spectral norm for matrices. We use $\mathrm{diag}(\mathbf{v}) \in \mathbb{R}^{m\times m}$ to denote the diagonal matrix that has $\mathbf{v} \in \mathbb{R}^m$ at its diagonal. We also use the shorthand notation $[m] \coloneqq \{1, \ldots, m\}$ and $\mathcal{B}(\mathbf{v}, r) \coloneqq \{\mathbf{x} \in \mathbb{R}^m \mid \|\mathbf{x} - \mathbf{v}\|_2 \le r\}$.

We say that a continuously differentiable function $f : \mathbb{R}^m \to \mathbb{R}^n$ is K-Lipschitz continuous if for all $\mathbf{x} \in \mathbb{R}^m$, we have $\|\nabla f(\mathbf{x})\| \le K$. We say that $f : \mathbb{R}^n \to \mathbb{R}$ is $\rho-$weakly convex over $\mathcal{B}(\mathbf{v}, r)$ if $f + \frac{\rho}{2}\|\cdot\|^2$ is convex over $\mathcal{B}(\mathbf{v}, r)$; it is $L-$smooth over $\mathcal{B}(\mathbf{v}, r)$ if for all $\mathbf{x_1}, \mathbf{x_2} \in \mathcal{B}(\mathbf{v}, r)$, we have

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \le L\|\mathbf{x}_1 - \mathbf{x}_2\|.$$

An $L-$smooth function over $\mathcal{B}(\mathbf{v}, r)$ is automatically $L-$weakly convex over $\mathcal{B}(\mathbf{v}, r)$. From now on, we will not distinguish between a Lipschitz smooth function and a function with a Lipschitz continuous gradient. Also, we will not differentiate among the Lipschitz smoothness constant, the Lipschitz continuity constant of the gradient, and the curvature.

2.1 Problem setup

RLVR. We adopt a sequence-level RL with verifiable reward formulation for LLM training. RLVR [12, 8, 28, 30] has recently gained attention as an effective approach for enhancing the reasoning performance of LLMs. Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ be the set of questions and $\mathcal{O} = \{o_1, \ldots, o_K\}$ be the set of possible output sequences. A predefined deterministic reward function $r : \mathcal{Q} \times \mathcal{O} \to \{0, 1\}$ evaluates whether the output $o \in O$ for a certain $q \in \mathcal{Q}$ is correct or not. Specifically, $r(q, o) = 1$ if the response is correct and $r(q, o) = 0$ otherwise. Given a question q an LLM generates the response $o \sim \pi_\theta(q)$ using a stochastic policy $\pi_\theta : \mathcal{Q} \to \Delta(\mathcal{O})$ parameterized by $\theta \in \Theta$. The goal of RLVR is to learn a policy that maximizes the expected reward:

$$J(\theta) \coloneqq \frac{1}{n}\sum_{i=1}^{n} J_i(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{o\sim\pi_\theta(\cdot|q_i)}[r(o, q_i)] \tag{1}$$

where $J_i(\theta) \coloneqq \mathbb{E}_{o\sim\pi_\theta(\cdot|q_i)}[r(o, q_i)]$ denotes the expect reward of policy π_θ for question q_i.

Remark 1. In the scenario of LLM alignment (e.g., RLHF), to mitigate over-optimization of the reward model, an additional KL penalty term is often added:

$$J_{\mathrm{KL}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{o\sim\pi_\theta(\cdot|q_i)}[r(o, q_i)] - \beta\mathrm{KL}(\pi_\theta\|\pi_{\text{ref}}),$$

where $\beta \ge 0$ is the regularization parameter and π_{ref} corresponds to the reference policy, which is often the model after the Supervised Fine-Tuning (SFT) stage [26]. For RLVR, recent studies [4, 10] have shown that the KL term can be ignored when other hyperparameters are carefully set, and $J_{\mathrm{KL}}(\theta)$ reduces exactly to $J(\theta)$ for $\beta = 0$.

In this paper, we analyze a simplified on-policy setting with *exact* parameter updates for randomly selected questions. Specifically, at each iteration, a question q_i is sampled uniformly from $\mathcal{Q}$, and the corresponding gradient $\nabla J_i(\theta)$ can be computed exactly.

REINFORCE. Before we introduce GRPO, we first review the classical policy gradient algorithm REINFORCE. REINFORCE [31] appears well-suited for LLM alignment as it efficiently estimates the gradient with a single query of the language and reward model. Furthermore, it does not require training a value model, making it computationally more efficient compared to PPO. Nevertheless, REINFORCE suffers from a large variance in its stochastic gradients [13]. In this paper, we focus on the *exact* setting where the full gradient is computed for randomly selected questions. In this context, all critic-free policy gradient methods, including REINFORCE, reduce to the vanilla policy gradient method, as shown in Algorithm 1.

Algorithm 1 Critic-free Policy Gradient Method

Input: learning rate $\eta > 0$, initial parameter θ_0.
for $t = 1$ to T **do**
  Select $i(t)$ uniformly at random from $\{1, \ldots, n\}$
  $\theta_t \leftarrow \theta_{t-1} + \eta \nabla J_{i(t)}(\theta_{t-1})$
end for
Return: final policy $\pi_{\theta_{T-1}}$

PPO. PPO [25] is an actor-critic RL algorithm that is widely used in LLM alignment [21, 23]. Unlike REINFORCE, a critic model is trained alongside the actor model to estimate the state-value function for the current policy. In particular, the objective function of PPO is given by:

$$J_{\text{PPO}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{o \sim \pi_{\theta_{\text{old}}}(\cdot|q_i)} \Big[\min \Big(\gamma_i(o) A_i(o), \text{clip}(\gamma_i(o), 1-\epsilon, 1+\epsilon) A_i(o) \Big) \Big], \tag{2}$$

where ϵ is the clipping parameter, and $\gamma_i(o) \coloneqq \frac{\pi_\theta(o|q_i)}{\pi_{\theta_{\text{old}}}(o|q_i)}$ is the importance ratio between the current policy π_θ and the old policy $\pi_{\theta_{\text{old}}}$. The advantage $A_i(o)$ is calculated using Generalized Advantage Estimation (GAE) with a learned value-function critic. The requirement of an additional critic model causes substantial computational overhead and memory demands for LLM training.

2.2 GRPO

GRPO, introduced in DeepSeek-Math [26] and DeepSeek-R1 [8], builds upon the computational efficiency of REINFORCE by eliminating the learned value-function critic but significantly enhances its effectiveness. It computes the group mean reward as a baseline, and normalizes each response's update by the within-prompt reward standard deviation. Under the sequence-level RL setting, the GRPO objective closely resembles the PPO objective (2):

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{o \sim \pi_{\theta_{\text{old}}}(\cdot|q_i)} \Big[\min \Big(\gamma_i(o) A_i(o), \text{clip}(\gamma_i(o), 1-\epsilon, 1+\epsilon) A_i(o) \Big) \Big], \tag{3}$$

differing only in the advantage term $A_i(o)$. Specifically, the advantage $A_i(o)$ is given by:

$$A_i(o) \coloneqq \frac{r(q_i, o) - \mathbb{E}_{o' \sim \pi_{\theta_{\text{old}}}(\cdot|q_i)} \left[r\left(q_i, o'\right) \right]}{\sqrt{\text{Var}_{o' \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \left[r\left(q_i, o'\right) \right]}},$$

where no critic model is required. Now we impose the assumption that each question in $\mathcal{Q}$ admits a unique correct answer in $\mathcal{O}$, an assumption that is commonly adopted in the theoretical analysis of RL algorithms [18, 19, 14]:

Assumption 1 (Unique correct answer)**.** For any $q \in \mathcal{Q}$, there exists a unique $o^*(q) \in \mathcal{O}$ such that $r(q, o^*(q)) = 1$.

Under Assumption 1, we use a_i to denote the index of correct answer for question $q_i \in \mathcal{Q}$:

$$r(q_i, o_j) = \begin{cases} 1, & \text{if} \quad j = a_i \\ 0, & \text{if} \quad j \neq a_i, \end{cases} \tag{4}$$

and use $\mathbf{r}_i \in \mathbb{R}^K$ to denote the reward vector for question q_i: $[\mathbf{r}_i]_j = r(q_i, o_j) \quad \forall j \in [K]$. We consider the on-policy scenario where $\pi_\theta = \pi_{\theta_{\text{old}}}$, and the reward function has a unique correct answer. Therefore, the importance ratio remains to be $\gamma_i(o) = 1$, and

$$A_i(o) = \frac{r(q_i, o) - \pi^*_\theta(i)}{\sqrt{\pi^*_\theta(i)\Big(1 - \pi^*_\theta(i)\Big)}}. \tag{5}$$

where $\pi^*_\theta(i) \coloneqq \pi_\theta(o_{a_i} \mid q_i)$ denotes the success probability of policy π_θ on question q_i. The GRPO objective can be further simplified as:

$$J_{\text{GRPO}}(\theta) = \frac{1}{n}\sum_{i=1}^n J^i_{\text{GRPO}}(\theta) \coloneqq \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{o\sim\pi_\theta}\Big[A_i(o)\Big] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{o\sim\pi_\theta}\left[\frac{r(q_i, o) - \pi^*_\theta(i)}{\sqrt{\pi^*_\theta(i)\Big(1 - \pi^*_\theta(i)\Big)}}\right].$$

We also denote $\pi_\theta(i) \in \mathbb{R}^K$ as the probability vector for π_θ in question i, that is, $[\pi_\theta(i)]_j \coloneqq \pi_\theta(o_j \mid q_i), \forall j \in [K]$. Under the *exact* parameter updates setting, the gradient of GRPO does not change when removing the baseline. We term such an algorithm as (on-policy) GRPO. Our key observation is that the variance normalization in GRPO implicitly implements an adaptive step size. In particular,

$$\begin{aligned}\nabla J^i_{\text{GRPO}}(\theta) &= \mathbb{E}_{o\sim\pi_\theta}[A_i(o)\nabla \ln \pi_\theta(o \mid q_i)] = \mathbb{E}_{o\sim\pi_\theta}\left[\frac{r(q_i, o)}{\sqrt{\pi^*_\theta(i)\Big(1 - \pi^*_\theta(i)\Big)}}\nabla \ln \pi_\theta(o \mid q_i)\right] \\ &= \frac{\mathbb{E}_{o\sim\pi_\theta}[r(q_i, o)\nabla \ln \pi_\theta(o \mid q_i)]}{\sqrt{\pi^*_\theta(i)\Big(1 - \pi^*_\theta(i)\Big)}} = \frac{\nabla J_i(\theta)}{\sqrt{\pi^*_\theta(i)\big(1 - \pi^*_\theta(i)\big)}}.\end{aligned} \tag{6}$$

for all $i \in [n]$. The first and last equalities follow from the policy gradient theorem [27]. The second equality holds because subtracting a constant baseline does not affect the gradient calculation. The third equality follows from the fact that $A_i(o)$ is treated as constant in the gradient propagation. The pseudo-code for on-policy GRPO in the *exact* setting is provided in Algorithm 2.

Algorithm 2 On-policy GRPO

Input: learning rate $\eta > 0$, initial parameter θ_0.
for $t = 1$ to T **do**
  Select $i(t)$ uniformly at random from $\{1, \ldots, n\}$
  $\theta_t \leftarrow \theta_{t-1} + \eta \frac{\nabla J_{i(t)}(\theta_{t-1})}{\sqrt{\pi^*_{\theta_{t-1}}(i(t))\Big(1-\pi^*_{\theta_{t-1}}(i(t))\Big)}}$
end for
Return: final policy $\pi_{\theta_{T-1}}$

Policy parametrization and updates. In this paper, we focus on the *log-linear policy parametrization* [1, 34]. Specifically, we assume that for each question-output pair (q_i, o_j), there exists a constant feature vector $\mathbf{x}_{i,j} \in \mathbb{R}^d$ and the policy is given by:

$$\pi_\theta(o_j \mid q_i) \coloneqq \frac{\exp(\mathbf{x}_{i,j}^\top \theta)}{\sum_{l=1}^K \exp(\mathbf{x}_{i,l}^\top \theta)}. \tag{7}$$

We denote $X_i \in \mathbb{R}^{K\times d}$ as the feature matrix for question q_i: $X_i \coloneqq (\mathbf{x}_{i,1}, \cdots, \mathbf{x}_{i,K})^\top$. For ease of notation, we simply drop t from $i(t)$ whenever it clear in context. The update of Algorithm 1 takes the following form [14]:

$$\theta_t \leftarrow \theta_{t-1} + \eta X_i^\top \Big(\text{diag}\left(\pi_{\theta_{t-1}}(i)\right) - \pi_{\theta_{t-1}}(i)\pi_{\theta_{t-1}}^\top(i)\Big)\mathbf{r}_i. \tag{8}$$

Under Assumption 1, the update of REINFORCE (Algorithm 1) can be simplified as:

$$\theta_t \leftarrow \theta_{t-1} + \eta\Big(\pi^*_{\theta_{t-1}}(i)(1-\pi^*_{\theta_{t-1}}(i)\mathbf{x}_{i,a_i} - \pi^*_{\theta_{t-1}}(i)\sum_{j\neq a_i}[\pi_{\theta_{t-1}}(i)]_j \cdot \mathbf{x}_{i,j}\Big). \tag{9}$$

Similarly, the update of GRPO (Algorithm 2) can be simplified as:

$$\theta_t \leftarrow \theta_{t-1} + \eta\Big(\sqrt{\pi^*_{\theta_{t-1}}(i)(1-\pi^*_{\theta_{t-1}}(i))}\mathbf{x}_{i,a_i} - \sqrt{\frac{\pi^*_{\theta_{t-1}}(i)}{1-\pi^*_{\theta_{t-1}}(i)}}\sum_{j\neq a_i}[\pi_{\theta_{t-1}}(i)]_j \cdot \mathbf{x}_{i,j}\Big). \tag{10}$$

3 Theoretical results

In this section, we provide the convergence analysis for REINFORCE-style PG methods and GRPO in the *exact* setting. We show that GRPO achieves provably faster convergence than unnormalized REINFORCE. Note that PG methods with linear function approximation may fail to converge to the optimal policy [14]. Therefore, our analysis focuses on the convergence rate toward stationary points.

3.1 Local smoothness of objective function

We begin by relating reward variance to the smoothness of the objective. Let $X_{\max} = \max_{i\in[n]} \|X_i\|$.

Lemma 1. Under Assumption 1, for all $i \in [n]$ and $\theta \in \mathbb{R}^d$,

$$\|\nabla^2 J_i(\theta)\| \leq 4X^2_{\max} \cdot \pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big) = 4X^2_{\max} \cdot \mathrm{Var}(\pi_\theta(i)). \tag{11}$$

The proof is provided in Appendix A.1. This result shows that the local smoothness constant of $J_i(\theta)$ is proportional to the reward variance on q_i under policy π_θ.

Corollary 1. Under Assumption 1, for all $i \in [n]$ and $\theta \in \mathbb{R}^d$,

$$\|\nabla^2 J_i(\theta)\| \leq X^2_{\max}, \tag{12}$$

so that $J_i(\theta)$ is $X^2_{\max}$-smooth on $\mathbb{R}^d$.

For deterministic gradient descent on an L-smooth function, a step size of $1/L$ is a standard choice [6]. By Lemma 1, it is thus natural to adaptively adjust the step size according to the local smoothness constant of each $J_i(\theta)$, which varies across questions. GRPO achieves exactly this: variance normalization implicitly implements an adaptive step size matched to the local curvature of each prompt, providing a key explanation for its advantage over REINFORCE.

3.2 Orthogonal representation assumption

To extend this intuition from per-question objectives J_i to the averaged objective $J = \frac{1}{n}\sum_i J_i$, we must also control the interaction between different questions. In particular, we need to ensure that gradients for different prompts do not interfere destructively. Empirically, we observe in Section 4.1 that gradients associated with different questions are nearly orthogonal. Motivated by this, we adopt the following assumptions to facilitate analysis:

Assumption 2 (Orthogonal representation)**.** For all $i, j \in [n]$ with $i \neq j$, we have $X_i^\top X_j = \mathbf{0}$.

This assumption guarantees that the gradients associated with different questions are orthogonal, simplifying the analysis of convergence for both REINFORCE and GRPO.

To show the convergence guarantee for GRPO, we further impose the following assumption on the bound of within-prompt Bernoulli variance at every step:

Assumption 3 (Bounded variance)**.** For each $i \in [n]$, there exists a positive sequence $\{C_i(t)\}_{t=1}^\infty$

$$\sqrt{\pi^*_{\theta_t}(i)\big(1-\pi^*_{\theta_t}(i)\big)} \leq C_i(t) \leq \frac{1}{2}.$$

3.3 Convergence result

We established that the local smoothness constant of each question can be estimated through the variance of rewards. Before presenting the convergence result for Algorithm 1 and 2, we further characterize the local smoothness constant at each iteration via the following lemmas:

Lemma 2. Under Assumption 1, for all $i \in [n]$, $J_i(\theta)$ is $\frac{1}{2}X_{\max}-$Lipschitz over $\mathbb{R}^d$.

Lemma 3 (Non-uniform local smoothness)**.** Under Assumption 1, for all $i \in [n]$ and $\theta \in \mathbb{R}^d$, $J_i(\theta)$ is $\frac{5}{2}X_{\max}^2 \cdot \sqrt{\pi_\theta^*(i)(1-\pi_\theta^*(i))}-$smooth over $\mathcal{B}(\theta, \frac{1}{X_{\max}} \cdot \sqrt{\pi_\theta^*(i)(1-\pi_\theta^*(i))})$.

The proofs are available in Appendix A.2 and A.3. We now present our two main theorems, which establish the convergence guarantees for Algorithm 1 and Algorithm 2, respectively. Their proofs are deferred in Appendix B:

Theorem 1 (Convergence rate of REINFORCE)**.** Under Assumption 1 and Assumption 2, with the step size $\eta = \frac{1}{X_{\max}^2}$, the following holds:

$$\mathbb{E}[J_i(\theta_t)] - \mathbb{E}[J_i(\theta_{t-1})] \;\le\; -\frac{1}{2nX_{\max}^2}\,\mathbb{E}[\|\nabla J_i(\theta_{t-1})\|^2].$$

Moreover, we have

$$\begin{aligned}
\sum_{t=1}^{T}\mathbb{E}[\|\nabla J_i(\theta_t)\|^2] \;&\le\; 2n(1-\pi_{\theta_0}^*(i))\,X_{\max}^2,\\
\min_{t\in[T]}\mathbb{E}[\|\nabla J_i(\theta_t)\|^2] \;&\le\; \frac{2n(1-\pi_{\theta_0}^*(i))\,X_{\max}^2}{T}.
\end{aligned}$$

Theorem 2 (Convergence rate of GRPO)**.** Under Assumption 1-3 with the step size $\eta = \frac{1}{2X_{\max}^2}$, we have

$$\mathbb{E}[J_i(\theta_t)] - \mathbb{E}[J_i(\theta_{t-1})] \;\le\; -\frac{3}{16nX_{\max}^2\,C_i(t)}\,\mathbb{E}[\|\nabla J_i(\theta_{t-1})\|^2],$$

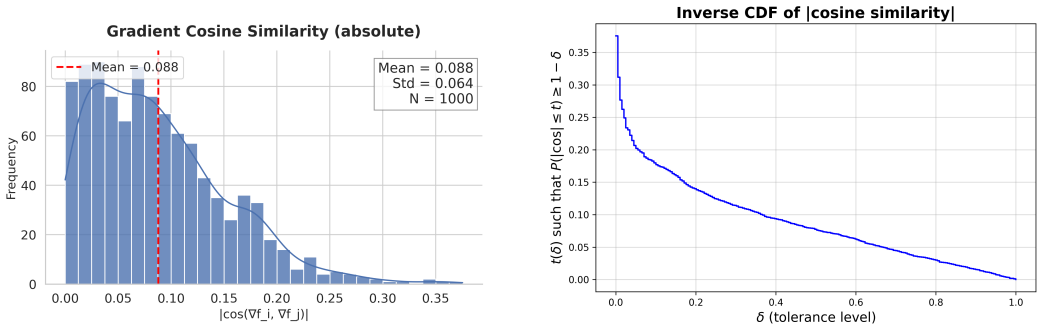
Moreover, we have

$$\begin{aligned}
\sum_{t=1}^{T}\mathbb{E}[\|\nabla J_i(\theta_t)\|^2] \;&\le\; 2n(1-\pi_{\theta_0}^*(i))X_{\max}^2 \cdot \frac{8}{3T}\sum_{t=0}^{T-1} C_i(t),\\
\min_{t\in[T]}\mathbb{E}[\|\nabla J_i(\theta_t)\|^2] \;&\le\; \frac{2n(1-\pi_{\theta_0}^*(i))X_{\max}^2}{T} \cdot \frac{8}{3T}\sum_{t=0}^{T-1} C_i(t).
\end{aligned}$$

According to Assumption 3, we use $\frac{1}{T}\sum_{i=0}^{T-1}\sqrt{\pi_{\theta_t}^*(i)\big(1-\pi_{\theta_t}^*(i)\big)}$ as an estimation of $\frac{1}{T}\sum_{t=0}^{T-1} C_i(t)$ in Theorem 2. Comparing Theorems 1 and 2, we observe that GRPO attains an average better convergence bound than the standard REINFORCE-style policy gradient methods, particularly if the constant factor

$$C(n,T) := \sum_{i=1}^{n}\sum_{j=0}^{T-1}\frac{8\sqrt{\pi_{\theta_j}^*(i)\big(1-\pi_{\theta_j}^*(i)\big)}}{3nT} = \sum_{i=1}^{n}\frac{8\sum_{j=0}^{T-1}\sqrt{\mathrm{Var}(\pi_{\theta_j})}}{3nT} < 1.$$

Here, $C(n,T)$ represents the average over prompts i and iterations j of the within-prompt Bernoulli standard deviation. $C(n,T)$ is typically much smaller than 1 when the question set contains diverse questions with varying levels of difficulty. A similar improvement can also be obtained in settings where the curvature varies across iterations. A more detailed discussion of the conditions under which $C(n,T) = o(1)$ is provided in Appendix C.



(a) Distribution of absolute cosine similarity ($|\cos|$)

(b) Inverse CDF of absolute cosine similarity ($|\cos|$)

Figure 1: Empirical validation of near-orthogonality assumption. (a) Histogram of absolute cosine similarities between question pairs. (b) Inverse CDF showing tail behavior.

4 Empirical studies

4.1 Validation of Orthogonal Assumption

A key assumption in our analysis is that representations of different training examples are nearly orthogonal in high-dimensional space. Formally, for two distinct questions $i \neq j$, we expect

$$\frac{\langle v_i, v_j \rangle}{\|v_i\|\|v_j\|} \approx 0, \tag{13}$$

where v_i denotes the representation vector (e.g., penultimate-layer hidden state) of question i. This assumption simplifies the analysis by ensuring cross-question interference is negligible.

We validate Assumption 2 on GSM8K [5] using `Qwen2.5-MATH-1.5B` [32]. For 1,000 random pairs of distinct questions, we extracted penultimate hidden states, pooled them into sentence-level embeddings, and measured absolute cosine similarities. As shown in Figure 1a, similarities are sharply concentrated near zero (mean ≈ 0.088, std ≈ 0.064). The inverse CDF in Figure 1b further shows that over 90% of pairs have similarity below 0.15, supporting the orthogonality assumption.

4.2 Validation of Local Curvature-Variance Connection

Table 1: Temporal Independence of Fisher Information and Reward Variance

Time Lag	**Mean Correlation**	**Significant** ($p < 0.05$)
Same time ($\Delta t = 0$)	0.342	Yes (0.008)
Different times ($\Delta t \neq 0$)	-0.028	No (0.18)

In our implementation, we compute the Fisher Information matrix following the efficient estimator proposed by [15]. Given a batch of prompts $\{q_i\}_{i=1}^{B}$ at iteration t, we: 1. Sample responses $\hat{o}_i \sim \pi_{\theta_t}(\cdot|q_i)$ for each prompt q_i 2. Compute the mini-batch gradient: $\nabla\hat{\mathcal{L}}_B(\theta_t) = \frac{1}{B}\sum_{i=1}^{B} \nabla \log \pi_{\theta_t}(\hat{o}_i|q_i)$ 3. Estimate the diagonal Fisher Information using the efficient estimator: $\mathbf{h}(\theta_t) = \text{diag}(\hat{F}_{\text{eff}}(\theta_t)) = B \cdot \nabla\hat{\mathcal{L}}_B(\theta_t) \odot \nabla\hat{\mathcal{L}}_B(\theta_t)$, where this estimator remains unbiased: $\mathbb{E}_{\hat{o}}[\text{diag}(\hat{F}_{\text{eff}}(\theta))] = \mathbb{E}_{\hat{o}}[\text{diag}(\hat{F}(\theta))]$(the expectation is taken over the sampled responses). The resulting Fisher Information $\mathbf{h}(\theta_t)$ serves as our curvature proxy, capturing the local smoothness of the loss landscape. In Table 1, prompt-level Fisher entries correlate with reward-variance at the same iteration (mean Pearson ≈ 0.34, $p < 0.01$) but not across different times, indicating the curvature–variance link is local in time and supports the result shown in Lemma 1.

Figure 2: Training reward standard deviation on GSM8K-easy. Normalized GRPO (**standard**, orange) rapidly reduces reward variance around $\sim$100 iterations and remains near zero, while **no_std** (blue) exhibits larger, persistent fluctuations. The reduced variance increases the signal-to-noise ratio of policy updates, leading to faster and smoother accuracy gains.

4.3 Comparisons on LLM Reasoning Task

Building upon the theoretical foundations established earlier, we conduct empirical evaluations to validate the effectiveness of different *advantage normalization* strategies in GRPO. Our experiments compare two normalization approaches across varying dataset difficulties on the GSM8K mathematical reasoning benchmark.

Experimental setup. We employ the `Qwen2.5-Math-1.5B` model as our base model, enhanced with Low-Rank Adaptation (LoRA) for parameter-efficient fine-tuning.

To study the effect of task difficulty, we partition the GSM8K training set by solution complexity into two splits: *Easy* (4,695 examples), and *Hard* (1,909 examples). We employ `Qwen2-7B-Instruct` as an evaluator to partition the dataset into distinct difficulty levels, thereby enabling a controlled study of how normalization behaves under varying difficulty regimes.

Normalization strategies. We evaluate three group-level (per-question) normalization approaches:

- Standard GRPO ($\mathcal{N}_{\text{standard}}$): per-question z-score normalization: $\hat{A}_{i,t} = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$
- No-Std ($\mathcal{N}_{\text{no-std}}$): mean-centering without variance scaling: $\hat{A}_{i,t} = r_i - \text{mean}(\mathbf{r})$.

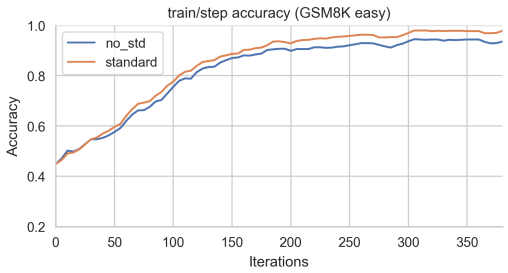
Evaluation metrics. We report complementary metrics: *sample accuracy* which is fraction of correct solutions among all generations.

Results and Discussion. According to Figure 2, both runs begin with a reward standard deviation near 0.55. The GRPO (normalized) run decreases earlier and more rapidly. The No normalization run declines more gradually and plateaus at a higher level. Overall, the normalized curve maintains a consistently lower variance throughout mid-to-late training phase.

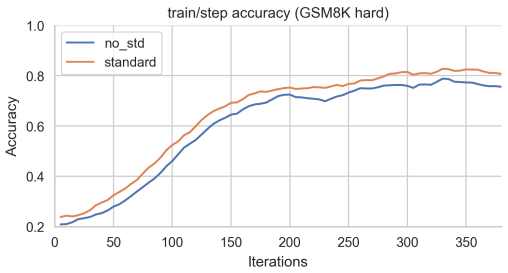
Across difficulties shown in Figure 3, we observe a clear variance-dependent pattern consistent with our theory:

- *Easy (low variance).* Both methods converge rapidly and saturate at high accuracy. GRPO Norm (standard) remains consistently but a little bit better, finishing around ≈ 0.98 versus ≈ 0.96 for No Normalization (no_std), i.e., a $\sim$2 points gap. After the initial rise, the two trajectories largely overlap and remain stable near the plateau.
- *Hard (high variance).* The benefit of normalization becomes more obvious once training moves beyond the mid range. GRPO Norm enters the 70–80% band earlier and retains a persistent lead, ending at about ≈ 0.81 compared to ≈ 0.78 for No Normalization (a $\sim$3 point gap). Variability is higher overall, but the normalized run shows a steadier climb in the later stages.

For easy questions, the initial accuracy starts near 50%. In this regime, GRPO provides little improvement over No-Std during the early iterations; however, the gap steadily widens after roughly 150–200 iterations, once accuracy is higher and variance is lower. For hard questions, the initial accuracy is much lower (around 20%), and GRPO yields a substantial early-phase acceleration. In both cases, the benefit of normalization grows as accuracy moves away from the 50% region, where



(a) GSM8K Easy: both settings rise quickly and saturate above 0.95; normalization stays $\sim$ 2 pts higher.



(b) GSM8K Hard: larger variance and a persistent gap; normalization finishes $\sim$3 pts higher.

Figure 3: Training accuracy vs. iterations on GSM8K Easy/Hard. Curves are smoothed with a 5-step moving window. *standard* (orange) uses normalization; *no_std* (blue) does not.

Bernoulli reward variance is highest, and becomes increasingly evident, especially in the later stages of training.

5 Conclusion

We showed that GRPO's normalization can be understood as an adaptive gradient mechanism, where reward variance controls the local curvature of the policy gradient and adjusts step sizes accordingly. This perspective explains its empirical advantages over unnormalized REINFORCE. Our theoretical results establish faster convergence under an orthogonal representation assumption, while experiments on synthetic tasks and GSM8K confirm that normalization improves stability and convergence, especially on harder questions with high reward variance. These findings provide a theoretical foundation for the success of GRPO and point to adaptive gradient mechanisms as a promising direction for designing robust critic-free RL algorithms for LLM training.

References

[1] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

[2] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

[3] Michael Bereket and Jure Leskovec. Uncalibrated reasoning: Grpo induces overconfidence for stochastic outcomes. *arXiv preprint arXiv:2508.11800*, 2025.

[4] Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025.

[5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[6] Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

[7] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

[8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[9] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

[10] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

[11] Nathan Lambert. Reinforcement learning from human feedback. *arXiv preprint arXiv:2504.12501*, 2025.

[12] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[13] Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

[14] Max Qiushi Lin, Jincheng Mei, Matin Aghaei, Michael Lu, Bo Dai, Alekh Agarwal, Dale Schuurmans, Csaba Szepesvari, and Sharan Vaswani. Rethinking the global convergence of softmax policy gradient with linear function approximation. *arXiv preprint arXiv:2505.03155*, 2025.

[15] Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023.

[16] Zhen Liu, Yuxuan Wang, Ziyang Chen, Han Wang, Jie Zhou, Maosong Sun, and et al. Formalmath: A large-scale formal benchmark for verifiable mathematical reasoning in lean4. *arXiv preprint arXiv:2505.02735*, 2025.

[17] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[18] Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International conference on machine learning*, pages 6820–6829. PMLR, 2020.

[19] Jincheng Mei, Zixin Zhong, Bo Dai, Alekh Agarwal, Csaba Szepesvari, and Dale Schuurmans. Stochastic gradient succeeds for bandits. In *International Conference on Machine Learning*, pages 24325–24360. PMLR, 2023.

[20] Youssef Mroueh, Nicolas Dupuis, Brian Belgodere, Apoorva Nitsure, Mattia Rigotti, Kristjan Greenewald, Jiri Navratil, Jerret Ross, and Jesus Rios. Revisiting group relative policy optimization: Insights into on-policy and off-policy training. *arXiv preprint arXiv:2505.22257*, 2025.

[21] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[22] Lei Pang and Ruinan Jin. On the theory and practice of grpo: A trajectory-corrected approach with fast convergence. *arXiv preprint arXiv:2508.02833*, 2025.

[23] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[24] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[25] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[26] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[27] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[28] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[29] Milan Vojnovic and Se-Young Yun. What is the alignment objective of grpo? *arXiv preprint arXiv:2502.18548*, 2025.

[30] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025.

[31] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[32] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[33] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[34] Rui Yuan, Simon Shaolei Du, Robert Mansel Gower, Alessandro Lazaric, and Lin Xiao. Linear convergence of natural policy gradient methods with log-linear policies. *ArXiv*, abs/2210.01400, 2022.

[35] Jiayi Zhang, Yuzhuo Tang, Haotian Li, Shengding Huang, Maosong Sun, Jie Zhou, and et al. Omni-math: A universal olympiad-level benchmark for mathematical reasoning. *arXiv preprint arXiv:2410.07985*, 2024.

A Analysis of the Local Smoothness Constant

A.1 Proof of Lemma 1

According to Lemma 17 in [14], for any $y \in \mathbb{R}^d$, we have

$$\mathbf{y}^\top \nabla^2 J_i(\theta)\mathbf{y} = \left(H\left(\pi_\theta(i)\right)\mathbf{r}_i\right)^\top \left(X_i\mathbf{y} \odot X_i\mathbf{y}\right) - 2\left(H\left(\pi_\theta(i)\right)\mathbf{r}_i\right)^\top \left(X_i\mathbf{y}\right)\left(\pi_\theta^\top(i)X_i\mathbf{y}\right)$$

where $H(\pi_\theta(i))$ is defined as $H\left(\pi_\theta\right) := \operatorname{diag}\left(\pi_\theta(i)\right) - \pi_\theta(i)\pi_\theta^\top(i) \in \mathbb{R}^{K\times K}$ and $\odot$ denotes the Hadamard (component-wise) product. Using the triangle inequality and Cauchy-Schwarz inequality, we get

$$\begin{aligned}
|\mathbf{y}^\top \nabla^2 J_i(\theta)\mathbf{y}| &\le |\left(H\left(\pi_\theta(i)\right)\mathbf{r}_i\right)^\top \left(X_i\mathbf{y} \odot X_i\mathbf{y}\right)| + 2|\left(H\left(\pi_\theta(i)\right)\mathbf{r}_i\right)^\top \left(X_i\mathbf{y}\right)| \cdot |\left(\pi_\theta^\top(i)X_i\mathbf{y}\right)| \\
&\le \|\left(H\left(\pi_\theta(i)\right)\mathbf{r}_i\right)\|_\infty \|X_i\mathbf{y} \odot X_i\mathbf{y}\|_1 + 2\|H\left(\pi_\theta(i)\right)\mathbf{r}_i\| \cdot \|X_i\mathbf{y}\| \cdot \|\pi_\theta(i)\| \cdot \|X_i\mathbf{y}\| \\
&= \|\left(H\left(\pi_\theta(i)\right)\mathbf{r}_i\right)\|_\infty \|X_i\mathbf{y}\|^2 + 2\|H\left(\pi_\theta(i)\right)\mathbf{r}_i\| \cdot \|\pi_\theta(i)\| \cdot \|X_i\mathbf{y}\|^2 \\
&\le \|\left(H\left(\pi_\theta(i)\right)\mathbf{r}_i\right)\|_\infty \|X_i\mathbf{y}\|^2 + 2\|H\left(\pi_\theta(i)\right)\mathbf{r}_i\| \cdot \|X_i\mathbf{y}\|^2.
\end{aligned} \tag{14}$$

The last inequality follows because $\|\pi_\theta(i)\| \le \|\pi_\theta(i)\|_1 = 1$. According to Assumption 1, we have

$$[H(\pi_\theta(i))\mathbf{r}_i]_j = \begin{cases} \pi_\theta^*(i)(1-\pi_\theta^*(i)), & \text{if } j = a_i \\ -\pi_\theta^*(i)[\pi_\theta(i)]_j, & \text{if } j \ne a_i \end{cases}$$

With this expression, we get

$$\|H(\pi_\theta(i))\mathbf{r}_i\|_\infty = \pi^*_\theta(i)(1-\pi^*_\theta(i)), \tag{15}$$

and

$$\begin{aligned}\|H(\pi_\theta(i))\mathbf{r}_i\| &= \pi^*_\theta(i)\sqrt{(1-\pi^*_\theta(i))^2 + \sum_{j\neq a_i}[\pi_\theta(i)]_j^2} \\ &\leq \pi^*_\theta(i)\sqrt{(1-\pi^*_\theta(i))^2 + \sum_{j\neq a_i}[\pi_\theta(i)]_j(1-\pi^*_\theta(i))} \\ &= \sqrt{2}\pi^*_\theta(i)(1-\pi^*_\theta(i)).\end{aligned} \tag{16}$$

Combining (15) and (16) with (14), we get

$$\begin{aligned}|\mathbf{y}^\top\nabla^2 J_i(\theta)\mathbf{y}| &\leq \|\left(H\left(\pi_\theta(i)\right)\mathbf{r}_i\right)\|_\infty \|X_i\mathbf{y}\|^2 + 2\|H\left(\pi_\theta(i)\right)\mathbf{r}_i\| \cdot \|X_i\mathbf{y}\|^2 \\ &\leq (2\sqrt{2}+1)\pi^*_\theta(i)(1-\pi^*_\theta(i))\|X_i\mathbf{y}\|^2 \\ &\leq (2\sqrt{2}+1)\pi^*_\theta(i)(1-\pi^*_\theta(i))\|X_i\|^2\|\mathbf{y}\|^2 \\ &\leq 4\pi^*_\theta(i)(1-\pi^*_\theta(i))X^2_{\max}\|\mathbf{y}\|^2\end{aligned}$$

where the third inequality is due to the definition of operator norm, and the last inequality is by definition of $X_{\max}$. Note that

$$\|\nabla^2 J_i(\theta)\| = \max_{\mathbf{y}} \frac{|\mathbf{y}^\top\nabla^2 J_i(\theta)\mathbf{y}|}{\|\mathbf{y}\|^2}$$

for symmetric Hessian matrix $\nabla^2 J_i(\theta)$, which completes the proof.

A.2 Proof of Lemma 2

According to (9), the gradient of $J_i(\theta)$ takes the following form:

$$\nabla J_i(\theta) = \mathbf{x}_{a_i}^\top(1-\pi^*_{\theta_t}(i))\pi^*_{\theta_t}(i) - \sum_{j\neq a_i}\mathbf{x}_j^\top\pi_{\theta_t}(i)_j \cdot \pi^*_{\theta_t}(i).$$

Note that a matrix's operator norm is larger than the norm of any of its row vector, we get

$$\begin{aligned}\|\nabla J_i(\theta)\| &\leq \|\mathbf{x}_{a_i}\|(1-\pi^*_{\theta_t}(i))\pi^*_{\theta_t}(i) + \sum_{j\neq a_i}\|\mathbf{x}_j\|\pi_{\theta_t}(i)_j \cdot \pi^*_{\theta_t}(i) \\ &\leq \|X_i\|(1-\pi^*_{\theta_t}(i))\pi^*_{\theta_t}(i) + \sum_{j\neq a_i}\|X_i\|\pi_{\theta_t}(i)_j \cdot \pi^*_{\theta_t}(i) \\ &= 2\|X_i\|(1-\pi^*_{\theta_t}(i))\pi^*_{\theta_t}(i) \\ &\leq \frac{1}{2}X_{\max}\end{aligned}$$

where the last inequality is due to the definition of $X_{\max}$, finishing the proof.

A.3 Proof of Lemma 3

By Assumption 1, the objective $J_i(\theta)$ is same as $\pi^*_\theta(i)$. From Lemma 2, $J_i(\theta)$ is $\frac{1}{2}X_{\max}$-Lipschitz. Consequently, for any $\theta' \in \mathcal{B}\Big(\theta, \frac{1}{X_{\max}}\sqrt{\pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big)}\Big)$, we have

$$\big|\pi^*_{\theta'}(i) - \pi^*_\theta(i)\big| \leq \tfrac{1}{2}X_{\max} \cdot \tfrac{1}{X_{\max}} \cdot \sqrt{\pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big)} = \tfrac{1}{2}\sqrt{\pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big)}.$$

Combining with Lemma 1,

$$\|\nabla^2 J_i(\theta')\| \leq \max_l 4X^2_{\max} \cdot l(1-l)$$

over $\mathcal{B}\Big(\theta, \frac{1}{X_{\max}}\sqrt{\pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big)}\Big)$, where l satisfies

$$\big|\, l - \pi^*_\theta(i)\,\big| \leq \tfrac{1}{2}\sqrt{\pi^*_\theta(i)\big(1-\pi^*_\theta(i)\big)}.$$

We denote $\pi_\theta^*(i)$ as a. Thus, proving Lemma 3 is equivalent as proving

$$f(a) \coloneqq \max_{l \in [a-\frac{\sqrt{a(1-a)}}{2}, a+\frac{\sqrt{a(1-a)}}{2}]} \frac{4l(1-l)}{\sqrt{a(1-a)}} \leq \frac{5}{2}.$$

WLOG, we assume $a \in [0, \frac{1}{2}]$ and consider two cases.

Case 1: When $a \in [\frac{1}{2} - \frac{\sqrt{5}}{10}, \frac{1}{2}]$, we know that

$$\frac{1}{2} \in [a - \frac{\sqrt{a(1-a)}}{2}, a + \frac{\sqrt{a(1-a)}}{2}],$$

which implies that

$$f(a) = \frac{1}{\sqrt{a(1-a)}} \leq f(\frac{1}{2} - \frac{\sqrt{5}}{2}) = \sqrt{5} \leq \frac{5}{2}.$$

Case 2: When $a \in [0, \frac{1}{2} - \frac{\sqrt{5}}{10}]$, we know that

$$\frac{1}{2} \notin [a - \frac{\sqrt{a(1-a)}}{2}, a + \frac{\sqrt{a(1-a)}}{2}],$$

which implies that

$$f(a) = \frac{(a + \frac{\sqrt{a(1-a)}}{2})(1 - a - \frac{\sqrt{a(1-a)}}{2})}{\sqrt{a(1-a)}} = 3\sqrt{a(1-a)} + (2 - 4a).$$

$f(a)$ takes its maximum when $a = \frac{1}{10}$ and $f(a) = \frac{5}{2}$.

Combining the above two cases, we conclude the lemma.

B Convergence Analysis of the Main Result

B.1 Auxiliary Lemma

Lemma 4. Under Assumption 1 and 2, for any $i, j \in [n], i \neq j$ and $\theta \in \mathbb{R}^d$, we have

$$\nabla J_i(\theta)^\top \nabla J_j(\theta) = 0 \tag{17}$$

Proof. According to (8), we get

$$\begin{aligned}
\nabla J_i(\theta)^\top \nabla J_j(\theta) &= \mathbf{r}_i^\top \left(\operatorname{diag}\left(\pi_\theta(i)\right) - \pi_\theta(i)\pi_\theta^\top(i)\right) X_i X_j^\top \left(\operatorname{diag}\left(\pi_\theta(j)\right) - \pi_\theta(j)\pi_\theta^\top(j)\right) \mathbf{r}_j \\
&= \mathbf{r}_i^\top \left(\operatorname{diag}\left(\pi_\theta(i)\right) - \pi_\theta(i)\pi_\theta^\top(i)\right) \mathbf{0} \left(\operatorname{diag}\left(\pi_\theta(j)\right) - \pi_\theta(j)\pi_\theta^\top(j)\right) \mathbf{r}_j \\
&= 0,
\end{aligned}$$

where the second step is by Assumption 2. □

B.2 Proof of Theorem 1

We consider a specific question q_l. Combining Lemma 4 with *log-linear policy parameterization* in our setting, if question $q_{i(t)}$ is selected on iteration t in REINFORCE, we get

$$\begin{aligned}
J_j(\theta_t) &= J_j(\theta_{t-1} + \eta \nabla J_i(\theta_{t-1})) \\
&= J_j(\theta_{t-1})
\end{aligned} \tag{18}$$

for any $i(t) \neq l$. That is, the parameter update on question $q_{i(t)}$ will not affect the expected reward on other questions.

If question $i(t) = l$ is selected on iteration t in REINFORCE, we have

$$\begin{aligned}
J_l(\theta_t) - J_l(\theta_{t-1}) &\geq \langle \theta_t - \theta_{t-1}, \nabla J_l(\theta_{t-1}) \rangle - \frac{X_{\max}^2}{2} \|\theta_t - \theta_{t-1}\|^2 \\
&= (\eta - \frac{X_{\max}^2}{2}\eta^2) \|\nabla J_l(\theta_{t-1})\|^2 \\
&= \frac{1}{2X_{\max}^2} \|\nabla J_l(\theta_{t-1})\|^2
\end{aligned} \tag{19}$$

where the first step is by Corollary 1, which also indicate that $J_i(\theta)$ is $X_{\max}^2$-weakly convex. Taking expectation of (19) on $i(t)$, we get

$$\mathbb{E}[J_l(\theta_t)] - \mathbb{E}[J_l(\theta_{t-1})] \geq \frac{1}{2nX_{\max}^2}\|\nabla J_l(\theta_{t-1})\|^2. \tag{20}$$

Summing up (20) for $t = 1, \ldots, T$, we get

$$\frac{1}{2nX_{\max}^2}\sum_{t=0}^{T-1}\mathbb{E}[\|J_l(\theta_{t-1})\|^2] \leq \mathbb{E}[J_l(\theta_T)] - J_l(\theta_0) \leq 1 - \pi_{\theta_0}^*(l).$$

This directly leads to

$$\min_{t\in\{0,1,\ldots,T-1\}}\mathbb{E}[\|\nabla J_l(\theta_t)\|^2] \leq \frac{2n(1-\pi_{\theta_0}^*(l))X_{\max}^2}{T}.$$

B.3 Proof of Theorem 2

Similar to (18) in the proof of Theorem 1, the gradient update based on question q_i does not affect the objective for question q_l if $i \neq l$. That is,

$$J_l(\theta_t) = \begin{cases} J_l(\theta_{t-1}), & \text{if } i(t) \neq l \\ J_l(\theta_t), & \text{if } i(t) = l. \end{cases} \tag{21}$$

Consider the case where $i(t) = l$, from the parameter update rule in GRPO, we get

$$\theta_t = \theta_{t-1} + \eta\Big(\sqrt{\pi_{\theta_{t-1}}^*(l)(1-\pi_{\theta_{t-1}}^*(l))}\mathbf{x}_{l,a_l} - \sqrt{\frac{\pi_{\theta_{t-1}}^*(l)}{1-\pi_{\theta_{t-1}}^*(l)}}\sum_{j\neq a_l}[\pi_{\theta_{t-1}}(l)]_j \cdot \mathbf{x}_{l,j}\Big).$$

Also, by setting $\eta = \frac{1}{2X_{\max}^2}$, we have

$$\begin{aligned}
&\|\eta\Big(\sqrt{\pi_{\theta_{t-1}}^*(l)(1-\pi_{\theta_{t-1}}^*(l))}\mathbf{x}_{l,a_l} - \sqrt{\frac{\pi_{\theta_{t-1}}^*(l)}{1-\pi_{\theta_{t-1}}^*(l)}}\sum_{j\neq a_l}[\pi_{\theta_{t-1}}(l)]_j \cdot \mathbf{x}_{l,j}\Big)\| \\
=&\frac{1}{2X_{\max}^2}\|\Big(\sqrt{\pi_{\theta_{t-1}}^*(l)(1-\pi_{\theta_{t-1}}^*(l))}\mathbf{x}_{l,a_l} - \sqrt{\frac{\pi_{\theta_{t-1}}^*(l)}{1-\pi_{\theta_{t-1}}^*(l)}}\sum_{j\neq a_l}[\pi_{\theta_{t-1}}(l)]_j \cdot \mathbf{x}_{l,j}\Big)\| \\
\leq&\frac{1}{2X_{\max}^2}\Big(\sqrt{\pi_{\theta_{t-1}}^*(l)(1-\pi_{\theta_{t-1}}^*(l))}\|\mathbf{x}_{l,a_l}\| + \sqrt{\frac{\pi_{\theta_{t-1}}^*(l)}{1-\pi_{\theta_{t-1}}^*(l)}}\sum_{j\neq a_l}[\pi_{\theta_{t-1}}(l)]_j \cdot \|\mathbf{x}_{l,j}\|\Big) \\
\leq&\frac{1}{2X_{\max}^2}(2\sqrt{\pi_{\theta_{t-1}}^*(l)(1-\pi_{\theta_{t-1}}^*(l))}X_{\max}) \\
=&\frac{1}{X_{\max}}\cdot\sqrt{\pi_{\theta_{t-1}}^*(l)(1-\pi_{\theta_{t-1}}^*(l))}.
\end{aligned}$$

This implies that $\theta_t \in \mathcal{B}(\theta, \frac{1}{X_{\max}}\cdot\sqrt{\pi_{\theta_{t-1}}^*(l)(1-\pi_{\theta_{t-1}}^*(l))})$. According to Lemma 3, we obtain

$$\begin{aligned}
J_l(\theta_t) &\geq J_l(\theta_{t-1}) + \langle\theta_t - \theta_{t-1}, \nabla J_l(\theta_{t-1})\rangle - \frac{5}{4}X_{\max}^2\cdot\sqrt{\pi_{\theta_{t-1}}^*(l)(1-\pi_{\theta_{t-1}}^*(l))}\|\theta_t - \theta_{t-1}\|^2 \\
&= J_l(\theta_{t-1}) + \frac{3}{16X_{\max}^2\sqrt{\pi_\theta^*(l)(1-\pi_\theta^*(l))}}\|\nabla J_l(\theta_{t-1})\|^2 \\
&\geq J_l(\theta_{t-1}) + \frac{3}{16X_{\max}^2 C_l(t-1)}\|\nabla J_l(\theta_{t-1})\|^2
\end{aligned} \tag{22}$$

where the last step is by Assumption 3. Taking expectation of (22) on $i(t)$, we have

$$\mathbb{E}[J_l(\theta_t)] \geq \mathbb{E}[J_l(\theta_{t-1})] + \frac{3}{16nX_{\max}^2 C_l(t-1)}\mathbb{E}[\|\nabla J(\theta_{t-1})\|^2]. \tag{23}$$

because the objective J_l remains unchanged if $i(t) \neq l$ according to (21). Summing up (23) for $t = 1, \ldots, T$, we get

$$\mathbb{E}[J_l(\theta_T)] \geq J_l(\theta_0) + \sum_{t=0}^{T-1} \frac{3}{16nX_{\max}^2 C_l(t-1)} \mathbb{E}[\|\nabla J(\theta_{t-1})\|^2]. \tag{24}$$

According to the Cauchy-Schwarz inequality, we obtain

$$\min_{t \in \{0,1,\ldots,T-1\}} \mathbb{E}[\|\nabla J_l(\theta_t)\|^2] \leq \frac{2n(1-\pi_{\theta_0}^*(l))X_{\max}^2}{T} \frac{8\sum_{t=0}^{T-1} C_l(t)}{3T}.$$

C Discussion on $C(n,T)$

We are interested in the meaningful small-constant regime where $C(n,T) = o(1)$. Let $\varepsilon_{i,j} := 1 - \pi_{\theta_j}(i) \in [0,1]$. Then

$$C(n,T) = \frac{8}{3nT} \sum_{i=1}^{n} \sum_{j=0}^{T-1} \sqrt{\pi_{\theta_j}(i)\big(1-\pi_{\theta_j}(i)\big)} \leq \frac{8}{3n} \sum_{i=1}^{n} \underbrace{\frac{1}{T} \sum_{j=0}^{T-1} \min\big\{\tfrac{1}{2}, \sqrt{\varepsilon_{i,j}}\big\}}_{=:A_i(T)}. \tag{25}$$

Hence $C(n,T) = o(1)$ whenever each prompt's Cesàro mean $A_i(T) \to 0$. A convenient pointwise bound is

$$0 \leq \sqrt{\pi(1-\pi)} \leq \min\Big\{\tfrac{1}{2}, \sqrt{1-\pi}\Big\}.$$

A sufficient (and essentially necessary) condition is that, for every fixed $\delta > 0$,

$$\frac{1}{T}\Big|\{\, j < T : \ \varepsilon_{i,j} \geq \delta \,\}\Big| \xrightarrow[T\to\infty]{} 0 \qquad \text{for all } i \in [n].$$

We provide improvement regimes under which $A_i(T) \to 0$ below.

(i) Exponential improvement. If $\varepsilon_{i,j} \leq c_i \rho_i^j$ with $\rho_i \in (0,1)$, then

$$\frac{1}{T} \sum_{j<T} \sqrt{\varepsilon_{i,j}} \leq \frac{\sqrt{c_i}}{T} \sum_{j<T} \rho_i^{j/2} = O\big(\tfrac{1}{T}\big),$$

so $C(n,T) = O(1/T) = o(1)$.

(ii) Polynomial improvement. If $\varepsilon_{i,j} \leq c_i\, j^{-\alpha_i}$ for some $\alpha_i > 0$, then

$$\frac{1}{T} \sum_{j<T} \sqrt{\varepsilon_{i,j}} \leq \frac{\sqrt{c_i}}{T} \sum_{j<T} j^{-\alpha_i/2} = \begin{cases} O\big(T^{-\alpha_i/2}\big), & 0 < \alpha_i < 2, \\ O\big((\log T)/T\big), & \alpha_i = 2, \\ O\big(1/T\big), & \alpha_i > 2, \end{cases}$$

hence $C(n,T) = o(1)$ for any $\alpha_i > 0$. A notable special case is the *harmonic* regime $\varepsilon_{i,j} = \Theta(1/j)$, which yields

$$C(n,T) = O\Big(\sqrt{\tfrac{\log T}{T}}\Big) = o(1).$$

Note. This refines the example following Eq. (14): the intended assumption is $1 - \pi_{\theta_j}(i) = \Theta(1/j)$ (not $\Theta(1/T)$).

Observe that

$$\sum_{j=0}^{T-1} \sqrt{\pi_{\theta_j}^*(i)(1-\pi_{\theta_j}^*(i))} \leq \sqrt{T \cdot \sum_{j=0}^{T-1} \pi_{\theta_j}^*(i)(1-\pi_{\theta_j}^*(i))} \leq \sqrt{T \cdot \sum_{j=0}^{T-1} (1-\pi_{\theta_j}^*(i))}.$$

For instance, if $(1-\pi_{\theta_j}^*(i)) = \mathcal{O}(1/T)$, then $C(n,T) = \mathcal{O}(\sqrt{\log T/T})$.

(iii) Log-slow improvement. If $\varepsilon_{i,j} \asymp 1/\log(j+e)$, then

$$\frac{1}{T}\sum_{j<T}\sqrt{\varepsilon_{i,j}} \asymp \frac{1}{\sqrt{\log T}}, \qquad \Rightarrow \qquad C(n,T) = O\big(1/\sqrt{\log T}\big) = o(1).$$

(iv) Persistent hard prompts (plateau). If for some i there exists $\varepsilon_0 > 0$ such that $\varepsilon_{i,j} \geq \varepsilon_0$ on a non-vanishing fraction of iterations, then $A_i(T)$ is bounded away from 0, and $C(n,T) \not\to 0$. Thus, *for fixed* n, every prompt must become (asymptotically) easy in Cesàro mean in order to have $C(n,T) = o(1)$.

(v) Mixed populations (curriculum/heterogeneity). Suppose the prompts split into $\mathcal{E}$ (easy) with $\varepsilon_{i,j} \to 0$ sufficiently fast (any of (i)–(iii)), and $\mathcal{H}$ (hard) with $\limsup_T A_i(T) \geq c > 0$. Then

$$C(n,T) \;\leq\; \frac{8}{3}\Big(\frac{|\mathcal{E}|}{n}\cdot o(1) \;+\; \frac{|\mathcal{H}|}{n}\cdot \Theta(1)\Big).$$

Therefore $C(n,T) = o(1)$ iff $|\mathcal{H}| = 0$ (for fixed n); if n grows with T, one additionally needs $|\mathcal{H}|/n \to 0$.

A universal upper bound. Since $\sqrt{\pi(1-\pi)} \leq \frac{1}{2}$, we always have

$$C(n,T) \leq \frac{8}{3nT}\cdot n \cdot \frac{T}{2} = \frac{4}{3}.$$

Thus the multiplicative factor in Eq. (14) is at worst a constant; the benefit over the unnormalized baseline is most pronounced when $C(n,T) = o(1)$, i.e., when success probabilities approach 1 on (almost) all prompts.

In summary, any training dynamic in which $\pi_{\theta_j}(i) \to 1$ for every prompt, no matter how slowly (even logarithmically), drives $C(n,T) \to 0$. Faster per-prompt improvement directly tightens Eq. (14), quantifying how GRPO's normalization converts heterogeneous per-prompt "curvature" into a vanishing multiplicative constant in the convergence bound.